

# Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP2005/017647

International filing date: 26 September 2005 (26.09.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP  
Number: 2004-283530  
Filing date: 29 September 2004 (29.09.2004)

Date of receipt at the International Bureau: 15 November 2005 (15.11.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland  
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日  
Date of Application: 2 0 0 4 年 9 月 2 9 日

出 願 番 号  
Application Number: 特 願 2 0 0 4 - 2 8 3 5 3 0

パリ条約による外国への出願  
に用いる優先権の主張の基礎  
となる出願の国コードと出願  
番号  
J P 2 0 0 4 - 2 8 3 5 3 0  
The country code and number  
of your priority application,  
to be used for filing abroad  
under the Paris Convention, is

出 願 人  
Applicant(s): ソニー株式会社

2 0 0 5 年 1 0 月 2 6 日

特許庁長官  
Commissioner,  
Japan Patent Office

中 嶋



【書類名】	特許願
【整理番号】	0490350703
【提出日】	平成16年 9月29日
【あて先】	特許庁長官殿
【国際特許分類】	G06F 9/46
【発明者】	
【住所又は居所】	東京都港区南青山二丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内
【氏名】	横田 大輔
【特許出願人】	
【識別番号】	000002185
【氏名又は名称】	ソニー株式会社
【代理人】	
【識別番号】	100093241
【弁理士】	
【氏名又は名称】	宮田 正昭
【電話番号】	03-5541-7577
【選任した代理人】	
【識別番号】	100101801
【弁理士】	
【氏名又は名称】	山田 英治
【電話番号】	03-5541-7577
【選任した代理人】	
【識別番号】	100086531
【弁理士】	
【氏名又は名称】	澤田 俊夫
【電話番号】	03-5541-7577
【手数料の表示】	
【予納台帳番号】	048747
【納付金額】	16,000円
【提出物件の目録】	
【物件名】	特許請求の範囲 1
【物件名】	明細書 1
【物件名】	図面 1
【物件名】	要約書 1
【包括委任状番号】	9904833

## 【書類名】 特許請求の範囲

### 【請求項 1】

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を実行する情報処理装置であり、

物理プロセッサが割り当てられない状態にある論理プロセッサ対応のコンテキストを、該論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行するコンテキスト管理部を有することを特徴とする情報処理装置。

### 【請求項 2】

前記情報処理装置は、

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付ける処理を実行する制御 OS と、

論理プロセッサの適用主体としての論理パーティションを対応付けられたゲスト OS とを有し、

前記制御 OS は、

ゲスト OS から制御 OS に対するシステムコールに基づいて、該ゲスト OS 対応の論理パーティションが適用する論理プロセッサに対応するコンテキストを、前記ゲスト OS 対応の論理パーティションアドレス空間にマップして保存する処理を実行する構成であることを特徴とする請求項 1 に記載の情報処理装置。

### 【請求項 3】

前記制御 OS は、

ゲスト OS から制御 OS に対するシステムコールに基づいて、前記ゲスト OS 対応の論理パーティションが適用する論理プロセッサを物理プロセッサへの対応付け処理対象としてのタイムシェアリング処理対象から除外し、前記論理プロセッサ対応のコンテキストを、前記論理パーティションアドレス空間にマップして保存する処理を実行する構成であることを特徴とする請求項 2 に記載の情報処理装置。

### 【請求項 4】

前記制御 OS は、

論理プロセッサを物理プロセッサの割り付け対象とする活動状態と、

論理プロセッサを物理プロセッサの割り付け対象としない非活動状態と、

の 2 状態の切り替え処理を実行する構成であり、

ゲスト OS から制御 OS に対するシステムコールに基づいて、該ゲスト OS 対応の論理プロセッサを非活動状態に設定して、該論理プロセッサ対応のコンテキストを前記論理パーティションアドレス空間にマップして保存する処理を実行する構成であることを特徴とする請求項 2 に記載の情報処理装置。

### 【請求項 5】

前記制御 OS は、

ゲスト OS から制御 OS に対するシステムコールに基づいて、非活動状態に設定された前記ゲスト OS 対応の論理プロセッサを活動状態に再設定して、該論理プロセッサを物理プロセッサの割り付け対象に戻す処理を実行する構成であることを特徴とする請求項 4 に記載の情報処理装置。

### 【請求項 6】

前記コンテキスト管理部は、

論理プロセッサ対応のコンテキストとして、論理プロセッサのレジスタ、I/Oポート、ローカルストレージを含むコンテキストの保存処理を実行する構成であることを特徴とする請求項 1 に記載の情報処理装置。

### 【請求項 7】

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を実行する情報処理装置におけるプロセス制御方法であり、

論理プロセッサを物理プロセッサの割り付け対象から除外する論理プロセッサスケジュー

ーリングステップと、

物理プロセッサの割り付け対象から除外した物理プロセッサ対応のコンテキストを、該論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行するコンテキスト保存ステップと、

を有することを特徴とするプロセス制御方法。

**【請求項 8】**

前記プロセス制御方法は、さらに、

ゲスト OS から制御 OS に対するシステムコール出力ステップを有し、

前記論理プロセッサスケジューリングステップは、

前記システムコールに基づいて、論理プロセッサを物理プロセッサの割り付け対象から除外する処理を実行し、

前記コンテキスト保存ステップは、

前記システムコールに基づいて、ゲスト OS 対応の論理パーティションが適用する論理プロセッサに対応するコンテキストを、前記ゲスト OS 対応の論理パーティションアドレス空間にマップして保存する処理を実行することを特徴とする請求項 7 に記載のプロセス制御方法。

**【請求項 9】**

前記制御 OS は、

論理プロセッサを物理プロセッサの割り付け対象とする活動状態と、

論理プロセッサを物理プロセッサの割り付け対象としない非活動状態と、

の 2 状態の切り替え処理を実行し、

ゲスト OS から制御 OS に対するシステムコールに基づいて、該ゲスト OS 対応の論理プロセッサを非活動状態に設定して、該論理プロセッサ対応のコンテキストを前記論理パーティションアドレス空間にマップして保存する処理を実行することを特徴とする請求項 8 に記載のプロセス制御方法。

**【請求項 10】**

前記制御 OS は、さらに、

ゲスト OS から制御 OS に対するシステムコールに基づいて、非活動状態に設定された前記ゲスト OS 対応の論理プロセッサを活動状態に再設定して、該論理プロセッサを物理プロセッサの割り付け対象に戻す処理を実行することを特徴とする請求項 9 に記載のプロセス制御方法。

**【請求項 11】**

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を行なう構成においてプロセス制御処理を実行するコンピュータ・プログラムであり、

論理プロセッサを物理プロセッサの割り付け対象から除外する論理プロセッサスケジューリングステップと、

物理プロセッサの割り付け対象から除外した物理プロセッサ対応のコンテキストを、該論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行するコンテキスト保存ステップと、

を有することを特徴とするコンピュータ・プログラム。

【書類名】 明細書

【発明の名称】 情報処理装置、プロセス制御方法、並びにコンピュータ・プログラム

【技術分野】

【0001】

本発明は、情報処理装置、プロセス制御方法、並びにコンピュータ・プログラムに関する。さらに詳細には、複数の論理プロセッサが情報処理装置中のリソースを共有して各種のデータ処理を実行する構成において、論理プロセッサ対応のリソースに対するアクセス性を向上させ、効率的なデータ処理を実現する情報処理装置、プロセス制御方法、並びにコンピュータ・プログラムに関する。

【背景技術】

【0002】

1つのシステム内に複数のオペレーティングシステム（OS）を搭載したマルチOSシステムにおいては、各OSはそれぞれ異なるプロセスが実行可能であり、システムで共通のハードウェア、すなわちCPUやメモリ等を時系列に順次切り替えて利用した処理が行なわれる。

【0003】

複数OSの各々の実行プロセス（タスク）のスケジューリングは、例えばパーティション管理ソフトウェアによって実行される。1つのシステムにOS（ $\alpha$ ）とOS（ $\beta$ ）の2つのオペレーティングシステムが並存する場合、OS（ $\alpha$ ）の処理をパーティションAとし、OS（ $\beta$ ）の処理をパーティションBとすると、パーティション管理ソフトウェアは、パーティションAとパーティションBの実行スケジュールを決定し、決定したスケジュールに基づいて、ハードウェア資源を割り当てて各OSにおける処理を実行する。

【0004】

マルチOS型のシステムにおけるタスク管理を開示した従来技術として、例えば、特許文献1がある。特許文献1には、複数のOSの各々において実行されるタスク管理において、緊急性の高い処理を優先的に処理させるためのタスクスケジューリング手法を開示している。

【0005】

前述したように、各種のデータ処理の実行主体はパーティションとして設定される。具体的には、システム内の資源分配を受ける主体としての論理パーティションが設定され、論理パーティションに対して、物理プロセッサユニットの使用時間や、仮想アドレス空間、さらにメモリ空間などの様々なリソースが配分され、配分されたリソースを適用した処理が実行される。論理パーティションには、いずれかの物理プロセッサに対応する論理的なプロセッサが設定され、論理プロセッサに基づくデータ処理が実行される。論理プロセッサと物理プロセッサは必ずしも1対1の関係ではなく、1つの論理プロセッサに複数の物理プロセッサが対応付けられる場合もあり、複数の論理プロセッサに1つの物理プロセッサが対応付けられる場合もある。

【0006】

論理プロセッサを適用した複数の処理が並列に実行される場合、物理プロセッサは、複数の論理プロセッサによってスケジューリングされて使用されることになる。すなわち、複数の論理プロセッサは、タイムシェアリングによる物理プロセッサの使用を行なうことになる。

【0007】

例えば図1に示すように、1つの物理プロセッサに対して1つの論理プロセッサが割り当てられている場合、すなわち、論理プロセッサ（ア）が物理プロセッサ（1）を占有して利用し、論理プロセッサ（イ）が物理プロセッサ（2）を占有して利用している場合の論理プロセッサ（ア）に対するアクセス処理について考察する。

【0008】

例えば論理プロセッサ（ア）を設定された論理パーティションに対応するOSが、論理プロセッサ（ア）に対してアクセスしようとする。論理プロセッサ（ア）の占有している

物理プロセッサ（１）が保持するローカルストレージ領域などは、占有して利用されている物理プロセッサ（１）に対応する領域として論理プロセッサ（ア）に対応する論理パーティションのアドレス空間にマップされているため、論理パーティションに対応するＯＳは、常に論理プロセッサ（ア）に対してアクセスすることが可能である。論理プロセッサ（ア）に対するアクセスにより、論理プロセッサ（ア）に対応するローカルストレージ情報などの各種の情報を取得することができる。

【０００９】

しかし、例えば図２に示すように、１つの物理プロセッサに対して複数の論理プロセッサが割り当てられ、タイムシェアリングによる処理を実行している場合は、図２に示すアクセスＡのように、論理プロセッサ（ア）が物理プロセッサ（１）を適用しているタイミングでは、上述の処理と同様、論理パーティションに対応するＯＳは、論理プロセッサ（ア）に対してアクセスすることが可能であるが、図２に示すアクセスＢのように、論理プロセッサ（ア）が物理プロセッサ（１）を適用していないタイミングでは、論理プロセッサ（ア）はいずれの物理プロセッサも適用しておらず、物理プロセッサに対するローカルストレージ領域などの情報は、論理プロセッサ（ア）に対応する論理パーティションのアドレス空間にマップされておらず、論理プロセッサ（ア）のアクセスは実行不可能となる。この場合、論理プロセッサ（ア）がタイムシェアリングによって物理プロセッサを利用する時間まで待機することが必要となり、データ処理の遅延の発生などの問題が生ずることになる。

【特許文献１】特開２００３－３４５６１２号公報

【発明の開示】

【発明が解決しようとする課題】

【００１０】

本発明は、上述の問題点に鑑みてなされたものであり、複数の論理プロセッサが情報処理装置中のリソースを共有して各種のデータ処理を実行する構成において、論理プロセッサ対応のリソースに対するアクセス性を向上させ、効率的なデータ処理を実現する情報処理装置、プロセス制御方法、並びにコンピュータ・プログラムを提供することを目的とする。

【課題を解決するための手段】

【００１１】

本発明の第１の側面は、

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を実行する情報処理装置であり、

物理プロセッサが割り当てられない状態にある論理プロセッサ対応のコンテキストを、該論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行するコンテキスト管理部を有することを特徴とする情報処理装置にある。

【００１２】

さらに、本発明の情報処理装置の一実施態様において、前記情報処理装置は、複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付ける処理を実行する制御ＯＳと、論理プロセッサの適用主体としての論理パーティションを対応付けられたゲストＯＳとを有し、前記制御ＯＳは、ゲストＯＳから制御ＯＳに対するシステムコールに基づいて、該ゲストＯＳ対応の論理パーティションが適用する論理プロセッサに対応するコンテキストを、前記ゲストＯＳ対応の論理パーティションアドレス空間にマップして保存する処理を実行する構成であることを特徴とする。

【００１３】

さらに、本発明の情報処理装置の一実施態様において、前記制御ＯＳは、ゲストＯＳから制御ＯＳに対するシステムコールに基づいて、前記ゲストＯＳ対応の論理パーティションが適用する論理プロセッサを物理プロセッサへの対応付け処理対象としてのタイムシェアリング処理対象から除外し、前記論理プロセッサ対応のコンテキストを、前記論理パー

パーティションアドレス空間にマップして保存する処理を実行する構成であることを特徴とする。

【0014】

さらに、本発明の情報処理装置の一実施態様において、前記制御OSは、論理プロセッサを物理プロセッサの割り付け対象とする活動状態と、論理プロセッサを物理プロセッサの割り付け対象としない非活動状態と、の2状態の切り替え処理を実行する構成であり、

ゲストOSから制御OSに対するシステムコールに基づいて、該ゲストOS対応の論理プロセッサを非活動状態に設定して、該論理プロセッサ対応のコンテキストを前記論理パーティションアドレス空間にマップして保存する処理を実行する構成であることを特徴とする。

【0015】

さらに、本発明の情報処理装置の一実施態様において、前記制御OSは、ゲストOSから制御OSに対するシステムコールに基づいて、非活動状態に設定された前記ゲストOS対応の論理プロセッサを活動状態に再設定して、該論理プロセッサを物理プロセッサの割り付け対象に戻す処理を実行する構成であることを特徴とする。

【0016】

さらに、本発明の情報処理装置の一実施態様において、前記コンテキスト管理部は、論理プロセッサ対応のコンテキストとして、論理プロセッサのレジスタ、I/Oポート、ローカルストレージを含むコンテキストの保存処理を実行する構成であることを特徴とする。

【0017】

さらに、本発明の第2の側面は、

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を実行する情報処理装置におけるプロセス制御方法であり、

論理プロセッサを物理プロセッサの割り付け対象から除外する論理プロセッサスケジューリングステップと、

物理プロセッサの割り付け対象から除外した物理プロセッサ対応のコンテキストを、該論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行するコンテキスト保存ステップと、

を有することを特徴とするプロセス制御方法にある。

【0018】

さらに、本発明のプロセス制御方法の一実施態様において、前記プロセス制御方法は、さらに、ゲストOSから制御OSに対するシステムコール出力ステップを有し、前記論理プロセッサスケジューリングステップは、前記システムコールに基づいて、論理プロセッサを物理プロセッサの割り付け対象から除外する処理を実行し、前記コンテキスト保存ステップは、前記システムコールに基づいて、ゲストOS対応の論理パーティションが適用する論理プロセッサに対応するコンテキストを、前記ゲストOS対応の論理パーティションアドレス空間にマップして保存する処理を実行することを特徴とする。

【0019】

さらに、本発明のプロセス制御方法の一実施態様において、前記制御OSは、論理プロセッサを物理プロセッサの割り付け対象とする活動状態と、論理プロセッサを物理プロセッサの割り付け対象としない非活動状態と、の2状態の切り替え処理を実行し、ゲストOSから制御OSに対するシステムコールに基づいて、該ゲストOS対応の論理プロセッサを非活動状態に設定して、該論理プロセッサ対応のコンテキストを前記論理パーティションアドレス空間にマップして保存する処理を実行することを特徴とする。

【0020】

さらに、本発明のプロセス制御方法の一実施態様において、前記制御OSは、さらに、ゲストOSから制御OSに対するシステムコールに基づいて、非活動状態に設定された前記ゲストOS対応の論理プロセッサを活動状態に再設定して、該論理プロセッサを物理プロセッサの割り付け対象に戻す処理を実行することを特徴とする。



#### 【0021】

さらに、本発明の第3の側面は、

複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を行なう構成においてプロセス制御処理を実行するコンピュータ・プログラムであり、

論理プロセッサを物理プロセッサの割り付け対象から除外する論理プロセッサスケジューリングステップと、

物理プロセッサの割り付け対象から除外した物理プロセッサ対応のコンテキストを、該論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行するコンテキスト保存ステップと、

を有することを特徴とするコンピュータ・プログラムにある。

#### 【0022】

なお、本発明のコンピュータ・プログラムは、例えば、様々なプログラム・コードを実行可能な汎用コンピュータ・システムに対して、コンピュータ可読な形式で提供する記憶媒体、通信媒体、例えば、CDやFD、MOなどの記憶媒体、あるいは、ネットワークなどの通信媒体によって提供可能なコンピュータ・プログラムである。このようなプログラムをコンピュータ可読な形式で提供することにより、コンピュータ・システム上でプログラムに応じた処理が実現される。

#### 【0023】

本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づく、より詳細な説明によって明らかになるであろう。なお、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

#### 【発明の効果】

#### 【0024】

本発明の構成によれば、複数の論理サブプロセッサをタイムシェアリングによって物理サブプロセッサに対応付けてデータ処理を実行する構成において、物理サブプロセッサの割り当て対象になっていない論理サブプロセッサのコンテキストを、論理サブプロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行する構成としたので、論理サブプロセッサが物理サブプロセッサに割り当てられていない状況において、該論理サブプロセッサのコンテキストの取得が可能となり、論理サブプロセッサに対するアクセス、プログラムロードなどの処理を、論理サブプロセッサが物理サブプロセッサへ割り当てられるタイミングを待機することなく実行することが可能となり、データ処理効率を飛躍的に向上させることが可能となる。

#### 【0025】

さらに、本発明の構成によれば、コンテキストテーブルには、レジスタの内容の他に、従来のコンテキストテーブルには含まれない、例えばローカルストレージ領域、I/Oポートの内容なども併せて格納する構成としたので、ゲストOSは、物理サブプロセッサの使用状態にない非活動状態に設定されている論理サブプロセッサに対応する各種の状態情報として、レジスタの内容、ローカルストレージの内容、I/Oポートの内容を参照し、これらの情報に従ったリソースアクセスによるデータ処理を実行することが可能となり、データ処理効率が向上する。

#### 【発明を実施するための最良の形態】

#### 【0026】

以下、図面を参照しながら、本発明の情報処理装置、プロセス制御方法、並びにコンピュータ・プログラムの詳細について説明する。

#### 【0027】

まず、図3を参照して、本発明の情報処理装置のハードウェア構成例について説明する。プロセッサモジュール101は、複数のプロセッサ(Processing Unit)から構成されたモジュールであり、ROM(Read Only Memory)104、HDD123等に記憶されている

プログラムに従って、オペレーティングシステム（OS：Operating System）、OS対応のアプリケーション・プログラムなど、各種プログラムに従ったデータ処理を実行する。プロセッサモジュール101の詳細については、さらに、後段で、図4を参照して説明する。

#### 【0028】

グラフィックエンジン102は、プロセッサモジュール101から入力される指示に従って、出力部122を構成する表示デバイスに画面出力するためのデータ生成、例えば3Dグラフィック描画処理を実行する。メインメモリ（DRAM）103には、プロセッサモジュール101において実行するプログラムやその実行において適宜変化するパラメータ等を格納する。これらはCPUバスなどから構成されるホストバス111により相互に接続されている。

#### 【0029】

ホストバス111は、ブリッジ105を介して、PCI（Peripheral Component Interconnect/Interface）バスなどの外部バス112に接続されている。ブリッジ105は、ホストバス111、外部バス112間、およびコントローラ106、メモリカード107、その他のデバイスとのデータ入出力制御を実行する。

#### 【0030】

入力部121は、キーボード、ポインティングデバイスなどのユーザにより操作される入力デバイスからの入力情報を入力する。出力部122は、液晶表示装置やCRT（Cathode Ray Tube）などの画像出力部とスピーカ等からなる音声出力部から構成される。

#### 【0031】

HDD（Hard Disk Drive）123は、ハードディスクを内蔵し、ハードディスクを駆動し、プロセッサモジュール101によって実行するプログラムや情報を記録または再生させる。

#### 【0032】

ドライブ124は、装着されている磁気ディスク、光ディスク、光磁気ディスク、または半導体メモリ等のリムーバブル記録媒体127に記録されているデータまたはプログラムを読み出して、そのデータまたはプログラムを、インタフェース113、外部バス112、ブリッジ105、およびホストバス111を介して接続されているメインメモリ（DRAM）103に供給する。

#### 【0033】

接続ポート125は、外部接続機器128を接続するポートであり、USB、IEEE1394等の接続部を持つ。接続ポート125は、インタフェース113、外部バス112、ブリッジ105、およびホストバス111を介してプロセッサモジュール101等に接続されている。通信部126は、ネットワークに接続され、プロセッサモジュール101や、HDD123等から提供されるデータの送信、外部からのデータ受信を実行する。

#### 【0034】

次に、プロセッサモジュールの構成例について、図4を参照して説明する。図4に示すように、プロセッサモジュール200は、複数のメインプロセッサからなるメインプロセッサグループ201、複数のサブプロセッサからなる複数のサブプロセッサグループ202～20nによって構成されている。それぞれにメモリコントローラ、2次キャッシュが設けられている。各プロセッサグループ201～20nの各々は例えば8つのプロセッサユニットを有し、クロスバスアーキテクチャ、あるいはパケット交換式ネットワークなどによって接続されている。メインプロセッサグループ201のメインプロセッサの指示のもとに、複数のサブプロセッサグループ202～20nの1以上のサブプロセッサが選択され、所定のプログラムが実行される。

#### 【0035】

なお、本発明の装置は複数の物理サブプロセッサを持ち、ソフトウェアが物理サブプロセッサをタイムシェアリングで多重化し、論理プロセッサをOSに提供する。サブプロセッサを制御する制御OSはメインプロセッサ上で動作する。なお、本発明の方式はメイン

-サブプロセッサのように主従関係がある装置でなくても応用でき、メイン-サブプロセッサのように主従関係にないマルチプロセッサマシンにおいても適用可能である。

#### 【0036】

各プロセッサグループに設置されたメモリフローコントローラは、図3に示すメインメモリ103とのデータ入出力制御を実行し、2次キャッシュは、各プロセッサグループにおける処理用データの記憶領域として利用される。

#### 【0037】

次に、図5を参照して、本発明の情報処理装置におけるオペレーティングシステム(OS)構成を説明する。本発明の情報処理装置は複数のオペレーティングシステム(OS)が共存するマルチOS構成を持つ。図5に示すように、論理的な階層構成を持つ複数のオペレーティングシステム(OS)を持つ。

#### 【0038】

図5に示すように、下位レイヤに制御OS301を有し、上位レイヤに複数のゲストOS302, 303、およびシステム制御OS304が設定される。制御OS301は、システム制御OS304とともに図3、図4を参照して説明したプロセッサモジュールにおいて実行する各プロセスの1つの実行単位としての論理パーティションを実現し、システム内のハードウェア資源(計算機資源としてのメインプロセッサ、サブプロセッサ、メモリ、デバイス等)を各論理パーティションに割り振る処理を実行する。

#### 【0039】

ゲストOS302, 303は、例えばゲームOSやWindows(登録商標)、Linux(登録商標)などの各種のOSであり、制御OS301の制御の下に動作するOSである。なお、図5には、ゲストOS302, 303の2つのゲストOSのみを示してあるが、ゲストOSは任意の数に設定することが可能である。

#### 【0040】

ゲストOS302, 303は、制御OS301およびシステム制御OS304によって設定された論理パーティション内で動作し、その論理パーティションに割り当てられたメインプロセッサ、サブプロセッサ、メモリ、デバイス等のハードウェア資源を適用して各種のデータ処理を実行する。

#### 【0041】

例えば、ゲストOS(a)302は、制御OS301およびシステム制御OS304によって設定された論理パーティション2に割り当てられたメインプロセッサ、サブプロセッサ、メモリ、デバイス等のハードウェア資源を適用して、ゲストOS(a)302対応のアプリケーション・プログラム305を実行する。また、ゲストOS(b)303は、論理パーティションnに割り当てられたメインプロセッサ、サブプロセッサ、メモリ、デバイス等のハードウェア資源を適用して、ゲストOS(b)303対応のアプリケーション・プログラム306を実行する。制御OS301は、ゲストOSの実行に必要なインタフェースとしてゲストOSプログラミングインタフェースを提供する。

#### 【0042】

システム制御OS304は、論理パーティション管理情報を含むシステム制御プログラム307を生成し、システム制御プログラム307に基づくシステムの動作制御を制御OS301とともに実行する。システム制御プログラム307は、システム制御プログラム・プログラミングインタフェースを用いてシステムのポリシーを制御するプログラムである。システム制御プログラム・プログラミングインタフェースは、制御OS301からシステム制御OS304に提供される。例えばリソース配分の上限值を設定するなど、プログラムによる柔軟なカスタマイズのための手段を提供するのがシステム制御プログラム307の役割である。

#### 【0043】

システム制御プログラム307はシステム制御プログラム・プログラミングインタフェースを用いてシステムの振る舞いを制御することができる。例えば、新しく論理パーティションを作成し、その論理パーティションで新しいゲストOSを起動することができる。

複数のゲストOSが動作するシステムでは、それらのゲストOSはシステム制御プログラムにあらかじめプログラムされた順序で起動されることになる。また、ゲストOSから提出された資源割り当て要求を制御OS301が受理する前に検査し、システムのポリシーに従って修正したり、要求そのものを拒否したりすることができる。これにより、特定のゲストOSだけが資源を独占することがないようにすることができる。このように、システムのポリシーをプログラムとして実現したものがシステム制御プログラムである。

#### 【0044】

制御OS301はシステム制御OS304のために特別な論理パーティション（図では論理パーティション1）を割り当てる。制御OS301は、ハイパバイザモードで動作する。ゲストOSはスーパバイザモードで動作する。システム制御OS、アプリケーション・プログラムはプロブレムモード（ユーザモード）で動作する。

#### 【0045】

論理パーティションはシステム内の資源分配を受ける主体である。たとえばメインメモリ103（図3参照）はいくつかの領域へ分割され、それぞれの領域の使用権が論理パーティションに対して与えられる。論理パーティションに分配される資源の種別には以下のものがある。

- a) 物理プロセッサユニット使用時間
- b) 仮想アドレス空間
- c) 論理パーティション内で動作するプログラムがアクセスできるメモリ
- d) 制御OSが論理パーティションの管理のために用いるメモリ
- e) イベントポート
- f) デバイスの使用権
- g) キャッシュパーティション
- h) バス使用権

#### 【0046】

前述したように、ゲストOSは論理パーティションの中で動作する。ゲストOSは論理パーティションに割り当てられた資源を独占して各種のデータ処理を実行する。多くの場合、システム上で動作する個々のゲストOS毎に1つの論理パーティションが作成される。各論理パーティションにはユニークな識別子が与えられる。システム制御OS304は、論理パーティション管理情報として生成するシステム制御プログラムに識別子を対応づけて管理する。

#### 【0047】

論理パーティションは、制御OS301およびシステム制御OS304によって生成される。生成直後の論理パーティションは何も資源を持たず、使用資源の制限も設定されていない。論理パーティションには活動状態と終了状態という2つの状態がある。生成直後の論理パーティションは活動状態にある。論理パーティション内で動作するゲストOSの要求に基づいて論理パーティションは終了状態に遷移し、論理パーティションに割り当てられている全ての論理プロセッサが停止する。

#### 【0048】

なお、論理プロセッサは、論理パーティションに割り当てられる論理的なプロセッサであり、いずれかの物理プロセッサ、すなわち、図4に示すプロセッサグループ内のプロセッサに対応する。ただし、論理プロセッサと物理プロセッサは必ずしも1対1の関係ではなく、1つの論理プロセッサに複数の物理プロセッサが対応付けられる場合もあり、複数の論理プロセッサに1つの物理プロセッサが対応付けられる場合もある。論理プロセッサと物理プロセッサの対応付けは、制御OS301が決定する。

#### 【0049】

制御OS301は、各論理パーティションが使用する資源の量を制限する機能を備えている。ゲストOS302、303がシステム制御OS304との通信を行うことなく割り当て／解放が行える資源については使用量の制限が可能となっている。

#### 【0050】

各論理パーティションは制御シグナルポートを備えている。このポートには論理パーティション間のデータ交換／共有に必要な様々な制御シグナルが到着する。制御シグナルの例を以下に挙げる。

- a) 論理パーティション間イベントポートの接続依頼
- b) 論理パーティション間メッセージチャネルの接続依頼
- c) 共有メモリ領域への接続依頼

#### 【0051】

各論理パーティションに到着した制御シグナルは制御シグナルポートでキューイングされる。キューの深さは、メモリ資源が許す範囲であれば、制限は無い。キューイングに必要なメモリ資源は制御シグナルを送った論理パーティションから確保される。このポートから制御シグナルを取り出すためには、ゲストOSプログラミングインタフェースを呼び出す。空の制御シグナルポートに制御シグナルが到着したときに、任意のイベントポートにイベントを送信することが可能である。イベントポートの指定はゲストOSプログラミングインタフェースを呼び出すことによって行える。

#### 【0052】

制御OSは論理パーティションに対し、物理サブプロセッサを抽象化した論理サブプロセッサをリソース（計算機資源）として与える。前述したように物理サブプロセッサと論理サブプロセッサは一对一に対応付けされていない上、数が同じである必要もない。これを実現するために、制御OSは必要に応じて一つの物理サブプロセッサを複数の論理サブプロセッサに対応付けることができるようになっている。

#### 【0053】

論理サブプロセッサの数が物理サブプロセッサの数より多い場合、制御OSは物理サブプロセッサをタイムシェアリングして処理する。このため論理サブプロセッサは、時間の経過をともなって動作停止や動作再開を繰り返す可能性がある。ゲストOSはこれらの状態の変化を観測することができる。

#### 【0054】

図6を参照して、物理プロセッサと論理プロセッサとの対応について説明する。図6には1つのメインプロセッサ401と、物理サブプロセッサ411～414の物理プロセッサ構成を示し、さらに、2つの物理サブプロセッサ、すなわち物理サブプロセッサ（2）と、物理サブプロセッサ（4）のタイムシェアリング処理によって動作する論理サブプロセッサのタイムシーケンスを示している。

#### 【0055】

図6の例では、物理サブプロセッサ（2）は、  
時間 t a 0 ～ t a 1：論理サブプロセッサ（ア）  
時間 t a 1 ～ t a 2：論理サブプロセッサ（イ）  
時間 t a 2 ～ t a 3：論理サブプロセッサ（ウ）  
時間 t a 3 ～           ：論理サブプロセッサ（ア）

のタイムシェアリングによって各論理サブプロセッサが割り当てられ、それぞれの割り当て時間において、各論理サブプロセッサは物理サブプロセッサ（2）を適用した処理を実行している。

#### 【0056】

また、物理サブプロセッサ（4）は、  
時間 t b 0 ～ t b 1：論理サブプロセッサ（イ）  
時間 t b 1 ～ t b 2：論理サブプロセッサ（ウ）  
時間 t b 2 ～ t b 3：論理サブプロセッサ（ア）  
時間 t b 3 ～           ：論理サブプロセッサ（イ）

のタイムシェアリングによって各論理サブプロセッサが割り当てられ、それぞれの割り当て時間において、各論理サブプロセッサは物理サブプロセッサ（4）を適用した処理を実行している。

#### 【0057】

各論理サブプロセッサがタイムシェアリングによって物理サブプロセッサを利用した処理を実行し、さらに次の割り当て期間において物理サブプロセッサを適用したデータ処理を再開するために、データ処理中断時のハードウェア状態等の状態情報を保持しておくことが必要となる。この状態情報には、図に示す各物理サブプロセッサに対応するローカルストレージの内容、I/Oポートの内容、レジスタの内容が含まれる。

#### 【0058】

論理サブプロセッサが物理サブプロセッサに割り当てられている期間は、その論理サブプロセッサに対応する論理パーティションアドレス空間内の領域に、論理サブプロセッサの状態を反映した物理サブプロセッサのI/Oポートの領域の一部とローカルストレージ領域がマップされることになり、他のプロセッサを介したアクセスができる。例えば論理サブプロセッサの対応付けられた論理パーティションの設定されたゲストOSが他のプロセッサを介してアクセスすることができる。

#### 【0059】

しかし、論理サブプロセッサが物理サブプロセッサに割り当てられていない期間は、その論理サブプロセッサに対応する論理パーティションアドレス空間内の領域に、論理サブプロセッサの状態を反映した物理サブプロセッサのI/Oポートの領域とローカルストレージ領域がマップされていないため、先に図2を参照して説明したように一般的には論理サブプロセッサへのアクセスができなくなる。しかし、本発明では、論理プロセッサが物理プロセッサを利用していない期間において論理プロセッサへのアクセスを実現するため、I/Oポートの領域とローカルストレージ領域情報を含むコンテキストテーブルを他プロセッサからの参照可能な状態に保持する構成としている。

#### 【0060】

以下、このコンテキスト保存構成の詳細について説明する。以下の説明において、論理サブプロセッサを利用したデータ処理を実行するのは、図5を参照して説明したゲストOSに割り当てられる論理パーティションである。論理パーティションに対応する論理サブプロセッサに対して、タイムシェアリングによる物理プロセッサの割り当ては制御OSによって実行される。

#### 【0061】

論理サブプロセッサにはゲストOSによって制御される活動状態／非活動状態と、制御OSによって制御される実行状態／実行可能状態がある。これらの組み合わせで、論理サブプロセッサには以下の3つの状態が存在することになる。

- (a) 活動状態&実行可能状態
- (b) 活動状態&実行状態
- (c) 非活動状態

#### 【0062】

活動状態と非活動状態の違いは、論理サブプロセッサが制御OSによるタイムシェアリングの対象、すなわち物理サブプロセッサの割り当て対象になるかどうかの違いである。活動状態は、論理サブプロセッサがタイムシェアリングの対象になっている状態、すなわち物理プロセッサの割り当て対象になっている状態である。論理サブプロセッサが活動状態にある場合、制御OSはタイムシェアリングした物理サブプロセッサを適宜その論理サブプロセッサに割り付ける。

#### 【0063】

非活動状態は論理サブプロセッサがタイムシェアリングの対象になっていない状態である。論理サブプロセッサがこの非活動状態にある場合、制御OSは時分割した物理サブプロセッサをその論理サブプロセッサに割り付けない。しかし、この場合において、論理サブプロセッサのコンテキストはコンテキストテーブルに保持される。制御OSがこのコンテキストを論理パーティションアドレス空間にマップすることで、他プロセッサからの非活動状態にある論理プロセッサに対するアクセスが可能となる。なお、ゲストOSは論理サブプロセッサの活動状態／非活動状態を制御することができる。

#### 【0064】

実行状態と実行可能状態の違いは、活動状態の論理サブプロセッサが物理サブプロセッサで実際に実行されているかどうかの違いである。論理サブプロセッサが物理サブプロセッサより多い場合、先に図2を参照して説明したように、論理サブプロセッサはタイムシェアリングされた物理サブプロセッサで実現されるので、論理サブプロセッサは常に物理サブプロセッサで実行されているわけではない。実行状態は、論理サブプロセッサが物理サブプロセッサで実際に実行されている瞬間の状態を指す。

#### 【0065】

一方、実行可能状態は、論理サブプロセッサが物理プロセッサの割り当て対象となっている（活動状態）が、実際に物理サブプロセッサで実行されていない瞬間の状態である。

#### 【0066】

これらの状態間の遷移、すなわち、実行状態と実行可能状態との遷移は、制御OSが起こす論理サブプロセッサのコンテキストスイッチによって発生する。なお、ゲストOSは、論理サブプロセッサの実行状態／実行可能状態を知ることができる。

#### 【0067】

論理サブプロセッサが活動状態で実行状態であれば、論理パーティションアドレス空間内の領域に、論理サブプロセッサの状態を反映した物理サブプロセッサのI/Oポートの領域とローカルストレージ領域がマップされる。従って、論理サブプロセッサに対するアクセスが可能な状態である。

#### 【0068】

さらに、本発明の情報処理装置においては、論理サブプロセッサが非活動状態に移行する際に、制御OSが論理サブプロセッサの状態情報としての論理サブプロセッサのコンテキストを保存し、コンテキストテーブルを論理サブプロセッサに対応するゲストOSの論理パーティションアドレス空間内の領域にマップする。この処理によって、ゲストOSは、自己の論理パーティションアドレス空間から非活動状態にある論理サブプロセッサのコンテキストテーブルのリソース情報を参照することが可能となり、リソース情報の読み取り、書き込み、更新などの処理を行なうことができる。

#### 【0069】

さらに、コンテキストテーブルには、レジスタの内容の他に、従来のコンテキストテーブルには含まれない、例えばローカルストレージの内容やI/Oポートの内容なども併せて格納する構成としている。この結果、ゲストOSは、物理プロセッサの使用状態にない非活動状態に設定されている論理プロセッサに対応するこれらの各種の状態情報に従ったリソースアクセスによるデータ処理を実行することが可能となり、データ処理効率が向上する。

#### 【0070】

図7を参照して、ゲストOSによる論理サブプロセッサ対応のリソースアクセス態様について説明する。ゲストOSは論理パーティションが設定され、論理パーティションには論理サブプロセッサが対応付けられている。リソースは、ゲストOSアクセス不可能リソース501と、ゲストOSアクセス可能リソース502に区分される。

#### 【0071】

ゲストOS対応の論理パーティションに対応付けられた論理サブプロセッサが活動状態でありかつ実行状態である場合は、物理サブプロセッサ510によるデータ処理実行状態にあり、この状態では、ゲストOSアクセス不可能リソース501に含まれる汎用レジスタ521、I/Oポートの一部522、さらに、ゲストOSアクセス可能リソース502に含まれるI/Oポートの一部523、ローカルストレージ524、メインメモリ525を適用したデータ処理が実行される。

#### 【0072】

物理サブプロセッサによるデータ処理実行状態にある場合は、ゲストOSの論理パーティションアドレス空間には、I/Oポートの一部523領域やローカルストレージ524領域等がマップされており、ゲストOSは、これらのリソースに対してアクセスすることができる。

### 【0073】

次に、論理サブプロセッサが非活動状態、すなわち、物理サブプロセッサの割り当て対象からはずれた状態に移行すると、論理サブプロセッサのメインメモリ525にあるコンテキストテーブル531に当該コンテキストが保存される。

### 【0074】

メインメモリ525に保存される論理サブプロセッサのコンテキストテーブル531は、論理サブプロセッサに対応するゲストOSの論理パーティションアドレス空間内の領域にマップされるので、ゲストOSによってアクセス可能となる。

### 【0075】

図8を参照して、論理サブプロセッサに対応するゲストOSの論理パーティションアドレス空間と、物理アドレス空間との対応について説明する。

### 【0076】

図8には、論理サブプロセッサに対応するゲストOSの論理パーティションアドレス空間560と、物理アドレス空間570を示してある。物理アドレス空間570はメインメモリに相当する物理空間である。ゲストOSは、ゲストOSの論理パーティションアドレス空間領域にマップされた物理アドレス空間のみをアクセスできる。

### 【0077】

ゲストOSの論理パーティション対応の論理サブプロセッサが、活動状態でありかつ実行状態ある場合、すなわち物理サブプロセッサによるデータ処理実行状態にある場合は、ゲストOSの論理パーティションアドレス空間560には、I/Oポートの一部の領域とローカルストレージ領域561がマップされており、ゲストOSは、これらのリソースに対してアクセスすることができる。

### 【0078】

さらに、ゲストOSの論理パーティション対応の論理サブプロセッサが、非活動状態にある場合、すなわち物理サブプロセッサの割り当て対象から除外された状態にある場合は、ゲストOSの論理パーティションアドレス空間560には、前記I/Oポートの一部の領域とローカルストレージ領域に、さらに、前記I/Oポートに含まれない関連I/Oポートの領域とレジスタを加えたコンテキストテーブル562がマップされ、ゲストOSは、これらのリソースに対してアクセスすることができることになる。

### 【0079】

図9を参照して、コンテキスト保存処理の詳細について説明する。図9には、コンテキスト保存処理を実行する制御OS610と、保存対象のコンテキスト対応の論理プロセッサによるデータ処理を実行する論理パーティションの設定されたゲストOS620を示してある。

### 【0080】

先に図8を参照して説明したように、コンテキストがゲストOSから参照可能な状態に保存されるのは、論理サブプロセッサが非活動状態に設定される場合である。すなわち論理サブプロセッサが物理サブプロセッサの割り当て対象から除外される場合である。

### 【0081】

図9において、ゲストOS620は、制御OS610のシステムコール処理部611に対して、コンテキストをゲストOSから参照可能な位置への保存処理を要求するシステムコールを出力する。制御OS610のシステムコール処理部611は、ゲストOS620からのシステムコールを受領すると、論理プロセッサスケジューリング処理部612に対して、論理サブプロセッサのスケジューリング変更要求を出力し、さらにコンテキスト管理部に論理プロセッサに対応するコンテキスト保存を依頼する。

### 【0082】

システムコール処理部611は、要求に応じて、論理サブプロセッサを活動状態から非活動状態に遷移させる。すなわち、ゲストOS620に設定された論理パーティションに対応する論理サブプロセッサを物理サブプロセッサの割り当て対象から除外する処理を行なう。この処理によって、論理サブプロセッサは非活動状態に設定される。



### 【0083】

さらに、論理プロセッサスケジューリング処理部612は、コンテキスト管理部613に非活動状態に遷移した論理サブプロセッサに対応するコンテキストの保存処理を依頼する。コンテキスト管理部613は、論理プロセッサスケジューリング処理部612によって論理サブプロセッサのコンテキストの保存/復元が依頼されると、コンテキストの保存/復元を実行する。また、この論理サブプロセッサが非活動状態に設定されたことを確認すると、コンテキストを保存したコンテキストテーブルを論理パーティションアドレス空間へマップするようにメインメモリ管理部614へ依頼を出す。保存するコンテキストには、論理プロセッサのローカルストレージの内容、I/Oポートの内容、レジスタの内容が含まれる。

### 【0084】

メインメモリ管理部614は、物理アドレス空間としてのメインメモリに保存するコンテキストをゲストOS620の論理パーティションアドレス空間領域にマップし、ゲストOS620から参照可能な状態に設定する。メモリ管理部614は、ゲストOS620に対して、ゲストOS620の論理パーティションアドレス空間におけるコンテキストアクセス用アドレスを通知する。

### 【0085】

ゲストOSは、通知されたアドレスに従って、コンテキストを参照することが可能となり、コンテキストに基づくリソース、すなわち先に説明したようにメインプロセッサから直接アクセスできるI/Oポートの領域とローカルストレージ領域に、さらに、レジスタの内容、メインプロセッサから直接アクセスできないI/Oポートの内容を取得することができ、これらのリソース情報の読み取り、書き込み処理が可能となる。

### 【0086】

図10を参照して、コンテキストの保存処理、および保存されたコンテキストを利用したリソースアクセス処理について説明する。

### 【0087】

図10に示す処理は、ゲストOSが自己の論理パーティションの活動状態にある論理プロセッサを、非活動状態に設定して論理パーティションアドレス空間領域にマップされたコンテキストに基づいてリソースアクセスを実行し、さらに、その後、論理サブプロセッサを活動状態に戻す処理シーケンスを示している。各ステップについて説明する。

### 【0088】

ステップS110は、ゲストOSによる論理サブプロセッサのリソースアクセスの開始処理である。なお、このゲストOSには、論理パーティションが設定され、設定された論理パーティションに対応する論理サブプロセッサのリソースへのアクセスを行うものである。

### 【0089】

ステップS110の処理は、図10の右側に示すステップS111～S113の処理によって構成される。まず、ステップS111において、ゲストOSは制御OSに対してシステムコールを実行する。このシステムコールは、論理サブプロセッサを特定し、その論理サブプロセッサ対応のコンテキストをゲストOSから参照可能な位置へ保存する処理を要求するシステムコールである。

### 【0090】

制御OSは、ゲストOSからのシステムコール受領に応じて、ステップS112、S113の処理を実行する。なお、図10において、太線枠で示すステップS112、S113、S132は制御OSの処理に相当する。

### 【0091】

制御OSは、ステップS112において、ゲストOSから要求のあった論理サブプロセッサをタイムシェアリングの対象から除外する。すなわち物理サブプロセッサの割り当て対象から除外し、非活動状態に設定する。さらに、ステップS113において、論理サブプロセッサ対応のコンテキストの保存を実行する。保存されるコンテキストは、ゲストOS

Sの論理パーティションアドレス空間領域にマップされ、ゲストOSから参照可能な状態に設定される。なお、保存するコンテキストには、I/Oポートの内容、ローカルストレージの内容、レジスタの内容が含まれる。

#### 【0092】

ステップS120において、ゲストOSは、自己の論理パーティションアドレス空間領域からコンテキストにアクセスし、コンテキスト情報の取得、書き換え処理などリソースアクセス処理を実行する。

#### 【0093】

ゲストOSによるリソースアクセスが終了すると、ゲストOSは、次に、ステップS130において、論理サブプロセッサのリソースアクセス完了処理を実行する。

#### 【0094】

ステップS130の処理は、図10の右側に示すステップS131～S132の処理によって構成される。まず、ステップS131において、ゲストOSは制御OSに対してシステムコールを実行する。このシステムコールは、論理プロセッサを特定し、その論理プロセッサをタイムシェアリングの候補に設定する要求処理、すなわち非活動時状態から活動状態への状態遷移を要求するシステムコールである。

#### 【0095】

制御OSは、ゲストOSからのシステムコールの受信に応じて、ステップS132において、ゲストOSから要求のあった論理サブプロセッサをタイムシェアリングの対象に戻す処理を実行する。すなわち物理サブプロセッサの割り当て対象に再設定し活動状態に設定する。

#### 【0096】

これらの処理によって、論理サブプロセッサは、再度、活動状態に設定されて物理プロセッサの割り当て対象となり、タイムシェアリングに対応するスケジュールに応じて物理サブプロセッサを利用したデータ処理が実行される。

#### 【0097】

このように、本発明においては、物理サブプロセッサを適用したデータ処理を実行していない論理サブプロセッサについて、そのコンテキストの参照を可能な構成としたので、論理サブプロセッサ対応のリソースアクセスを効率的に実行することができ、物理サブプロセッサの割り当てまでリソースアクセスを待機する必要がなく、効率的なデータ処理が実現される。

#### 【0098】

図11、図12を参照して、本発明を適用した場合の非活動状態にある論理サブプロセッサ対応のコンテキストを利用した処理例について説明する。

#### 【0099】

論理サブプロセッサを適用した複数の処理が並列に実行される場合、物理サブプロセッサは、複数の論理サブプロセッサによってスケジュールリングされて使用される。すなわち、複数の論理サブプロセッサは、タイムシェアリングによる物理サブプロセッサの使用を行なうことになる。

#### 【0100】

図11は、2つの物理サブプロセッサ(1)、物理サブプロセッサ(2)に対して複数の論理サブプロセッサ(ア)～(エ)が割り当てられ、タイムシェアリングによる処理を実行している処理例を示している。ここで、あるゲストOS対応の論理サブプロセッサ(ア)に対するアクセスを実行する場合、タイムシェアリングによって論理プロセッサ(ア)がいずれかの物理サブプロセッサ(1)または物理サブプロセッサ(2)に割り当てられ、データ処理が実行されている場合には、前述したように、I/Oポートの領域の一部やローカルストレージ領域が、ゲストOS対応の論理パーティションアドレス空間にマップされた状態にあるので、論理サブプロセッサ(ア)およびそのリソースに対するアクセスが可能となる。しかし、タイムシェアリングによって論理サブプロセッサ(ア)がいずれの物理サブプロセッサ(1)、(2)にも割り当てられておらず、データ処理が実行さ

れていない場合には、I/Oポートの領域の一部やローカルストレージ領域が、ゲストOS対応の論理パーティションアドレス空間にマップされていない状態となる。

#### 【0101】

ここで、本発明の構成では、ゲストOSから制御OSに対するシステムコールによって、論理サブプロセッサ（ア）を非活動状態、すなわちタイムシェアリング対象から除外して、コンテキストを保存し、コンテキストテーブルをゲストOS対応の論理パーティションアドレス空間にマップする構成とした。すなわち、図11に示す論理サブプロセッサ（ア）701がゲストOSの参照可能なコンテキストを保持する状態となる。

#### 【0102】

図11では、時刻t1～t2の間において、論理サブプロセッサ（ア）を非活動状態に設定され、論理サブプロセッサ（ア）のコンテキストがゲストOS対応の論理パーティションアドレス空間にマップされ、ゲストOSから参照可能な状態に設定されていることを示している。時刻t1において、ゲストOSから制御OSに対するシステムコールが出力されて論理サブプロセッサ（ア）が活動状態から非活動状態に設定されるとともに、コンテキストがゲストOS対応の論理パーティションアドレス空間にマップされる。また、時刻t2において、ゲストOSから制御OSに対するシステムコールが出力されて論理サブプロセッサ（ア）が非活動状態から活動状態に設定される。

#### 【0103】

従来は、論理サブプロセッサが物理サブプロセッサに割り当てられていない期間は、論理サブプロセッサのコンテキストは参照可能ではなく、従って、OSから論理サブプロセッサに対するアクセスはエラーとなってしまうていた。すなわち、論理サブプロセッサ（ア）が物理サブプロセッサ（1）または2に割り当てられる次のタイミングまで待機することが必要となっていた。本発明の構成では、論理サブプロセッサの物理サブプロセッサの割り当てタイミングを待機することなく、論理サブプロセッサに対するアクセスが可能となる。

#### 【0104】

図12は、論理サブプロセッサを非活動状態に設定し、コンテキスト参照可能な期間に、論理サブプロセッサに対してプログラムの転送、およびプログラム起動指示を与えて、次の物理サブプロセッサの割り当て時に物理サブプロセッサを適用したプログラムの実行を即座に行い得る構成例を示している。

#### 【0105】

時刻t1において、ゲストOSから制御OSに対するシステムコールによって、論理サブプロセッサ（ア）を非活動状態、すなわちタイムシェアリング対象から除外して、コンテキストを保存し、コンテキストをゲストOS対応の論理パーティションアドレス空間にマップする。図12に示す時刻t1～t2において、ゲストOSは、ゲストOS対応の論理パーティションアドレス空間からコンテキストを参照、すなわち論理サブプロセッサ（ア）701にアクセスすることができる。

#### 【0106】

ゲストOSは、時刻t1～t2の間において、論理サブプロセッサ（ア）に対するデータ処理プログラムの転送処理を実行し、さらに、起動指示を出力する。具体的には、コンテキストテーブルに記録された論理サブプロセッサ（ア）のローカルストレージ領域に対するプログラムデータのロード処理などを実行することになる。これらの処理の終了後に時刻t2において、ゲストOSは制御OSに対して、システムコールを出力し、論理サブプロセッサ（ア）が非活動状態から活動状態に設定される。論理サブプロセッサ（ア）は次のタイムシェアリングによる物理サブプロセッサの使用時に、論理サブプロセッサ（ア）のローカルストレージ領域にロードされたプログラムを即座に実行することが可能となる。

#### 【0107】

従来は、このような処理を実行しようとする場合、論理サブプロセッサが物理サブプロセッサに割り当てられていない期間は、論理サブプロセッサのコンテキストは参照可能で

はなく、従って、論理サブプロセッサのローカルストレージ領域に対するアクセスはエラーとなってしまうていた。すなわち、論理サブプロセッサ（ア）が物理サブプロセッサ（１）または物理サブプロセッサ（２）に割り当てられる次のタイミングまでプログラムロード処理などは実行することできず、プログラムの実行タイミングが遅延してしまい、データ処理の効率性が落ちてしまうという欠点があったが、本発明の構成を適用することにより、データ処理効率を向上させることができる。

#### 【０１０８】

以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、特許請求の範囲の欄を参酌すべきである。

#### 【０１０９】

なお、明細書中において説明した一連の処理はハードウェア、またはソフトウェア、あるいは両者の複合構成によって実行することが可能である。ソフトウェアによる処理を実行する場合は、処理シーケンスを記録したプログラムを、専用のハードウェアに組み込まれたコンピュータ内のメモリにインストールして実行させるか、あるいは、各種処理が実行可能な汎用コンピュータにプログラムをインストールして実行させることが可能である。

#### 【０１１０】

例えば、プログラムは記録媒体としてのハードディスクやＲＯＭ（Read Only Memory）に予め記録しておくことができる。あるいは、プログラムはフレキシブルディスク、ＣＤ－ＲＯＭ（Compact Disc Read Only Memory）、ＭＯ（Magneto optical）ディスク、ＤＶＤ（Digital Versatile Disc）、磁気ディスク、半導体メモリなどのリムーバブル記録媒体に、一時的あるいは永続的に格納（記録）しておくことができる。このようなリムーバブル記録媒体は、いわゆるパッケージソフトウェアとして提供することができる。

#### 【０１１１】

なお、プログラムは、上述したようなリムーバブル記録媒体からコンピュータにインストールする他、ダウンロードサイトから、コンピュータに無線転送したり、ＬＡＮ（Local Area Network）、インターネットといったネットワークを介して、コンピュータに有線で転送し、コンピュータでは、そのようにして転送されてくるプログラムを受信し、内蔵するハードディスク等の記録媒体にインストールすることができる。

#### 【０１１２】

なお、明細書に記載された各種の処理は、記載に従って時系列に実行されるのみならず、処理を実行する装置の処理能力あるいは必要に応じて並列的にあるいは個別に実行されてもよい。また、本明細書においてシステムとは、複数の装置の論理的集合構成であり、各構成の装置が同一筐体内にあるものには限らない。

#### 【産業上の利用可能性】

#### 【０１１３】

以上、説明したように、本発明の構成によれば、複数の論理サブプロセッサをタイムシェアリングによって物理サブプロセッサに対応付けてデータ処理を実行する構成において、物理サブプロセッサの非適用状態にある論理サブプロセッサ対応のコンテキストを、論理サブプロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行する構成としたので、論理サブプロセッサが物理サブプロセッサに割り当てられていない状況において、該論理サブプロセッサのコンテキストの取得が可能となり、論理サブプロセッサに対するアクセス、プログラムロードなどの処理を、論理サブプロセッサが物理サブプロセッサへ割り当てられるタイミングを待機することなく実行することが可能となり、データ処理効率を飛躍的に向上させることが可能となる。

#### 【０１１４】

さらに、本発明の構成によれば、コンテキストテーブルには、レジスタの内容の他に、従来のコンテキストテーブルには含まれない、例えばローカルストレージの内容やI/Oポートの内容なども併せて格納する構成としたので、ゲストOSは、物理サブプロセッサが割り当てられていない非活動状態に設定されている論理サブプロセッサに対応する各種の状態情報として、I/Oポートの内容、ローカルストレージの内容、レジスタの内容を参照し、これらの情報に従ったリソースアクセスによるデータ処理を実行することが可能となり、データ処理効率が向上する。

#### 【図面の簡単な説明】

##### 【0115】

【図1】 タイムシェアリングによる論理プロセッサと物理プロセッサとの対応付け処理について説明する図である。

【図2】 タイムシェアリングによる論理プロセッサと物理プロセッサとの対応付け処理について説明する図である。

【図3】 本発明の情報処理装置の構成例を示す図である。

【図4】 プロセッサモジュールの構成例を示す図である。

【図5】 本発明の情報処理装置のオペレーションシステム構成を説明する図である。

【図6】 論理プロセッサと物理プロセッサとの対応付け処理について説明する図である。

【図7】 ゲストOSによって参照可能な領域、および本発明におけるコンテキストテーブルの設定例について説明する図である。

【図8】 ゲストOS対応の論理パーティションアドレス空間と、ゲストOSによって参照可能な情報について説明する図である。

【図9】 本発明におけるゲストOSと制御OSとにおいて実行する処理について説明する図である。

【図10】 本発明におけるゲストOSと制御OSとにおいて実行する処理シーケンスについて説明するフロー図である。

【図11】 本発明を適用したタイムシェアリングによる論理プロセッサと物理プロセッサとの対応付けおよびコンテキスト設定処理について説明する図である。

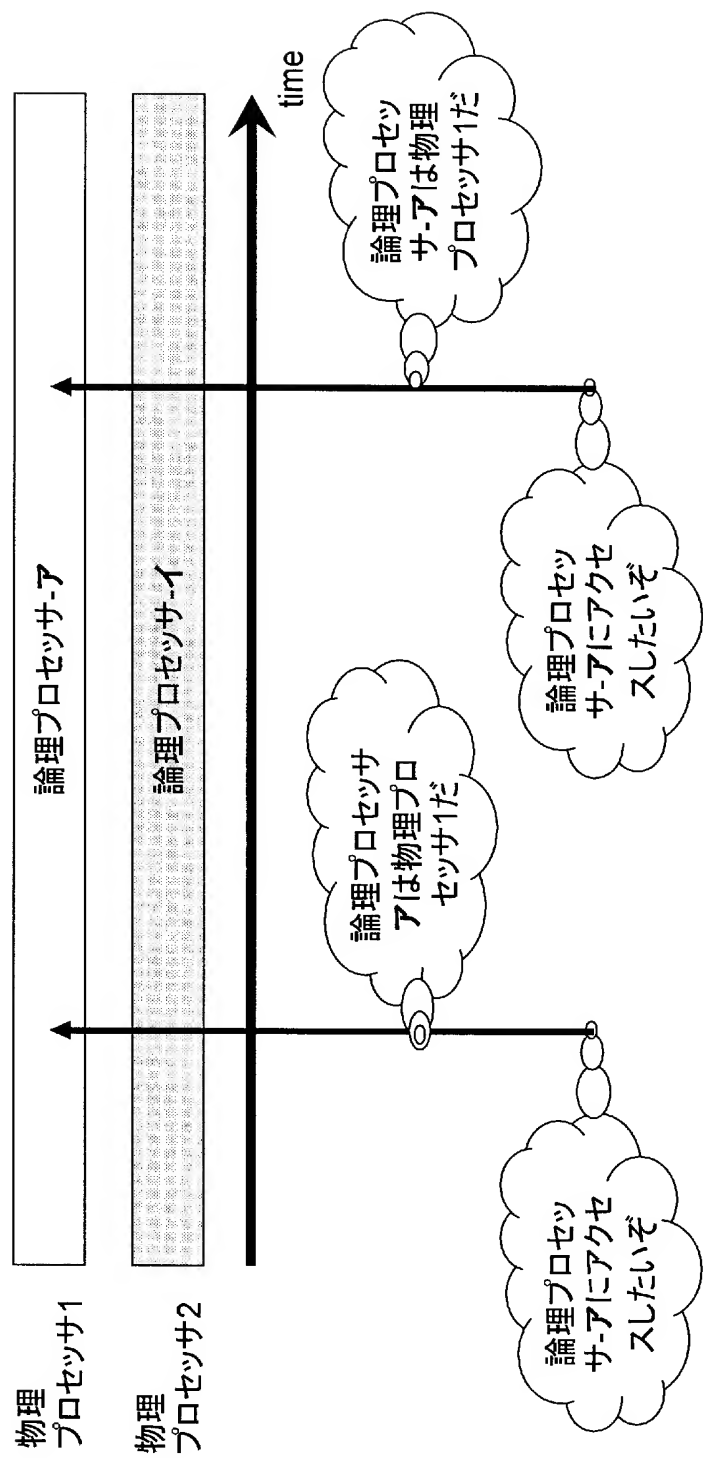
【図12】 本発明を適用したタイムシェアリングによる論理プロセッサと物理プロセッサとの対応付けおよびコンテキスト設定処理、プログラムロード処理について説明する図である。

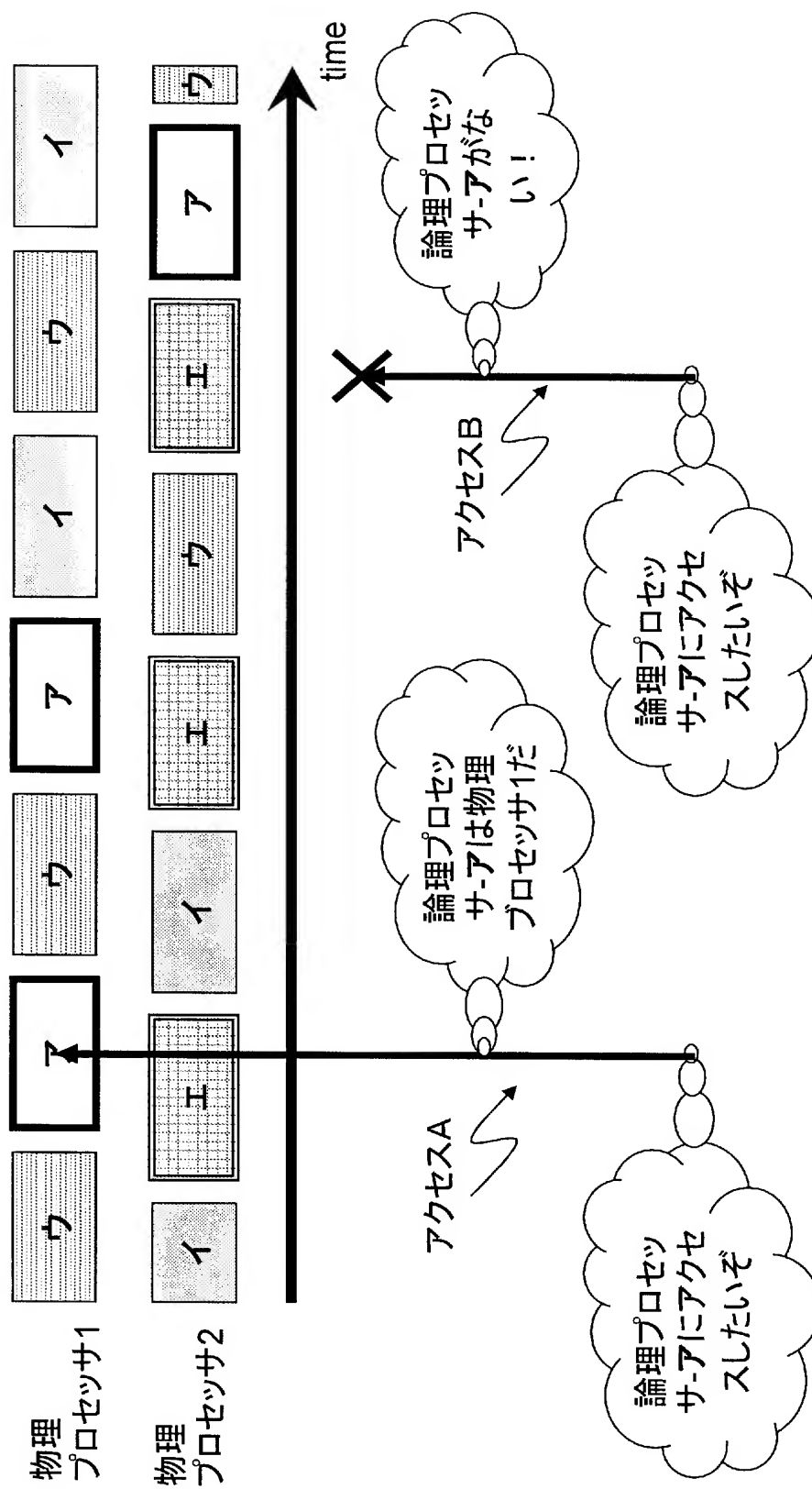
#### 【符号の説明】

##### 【0116】

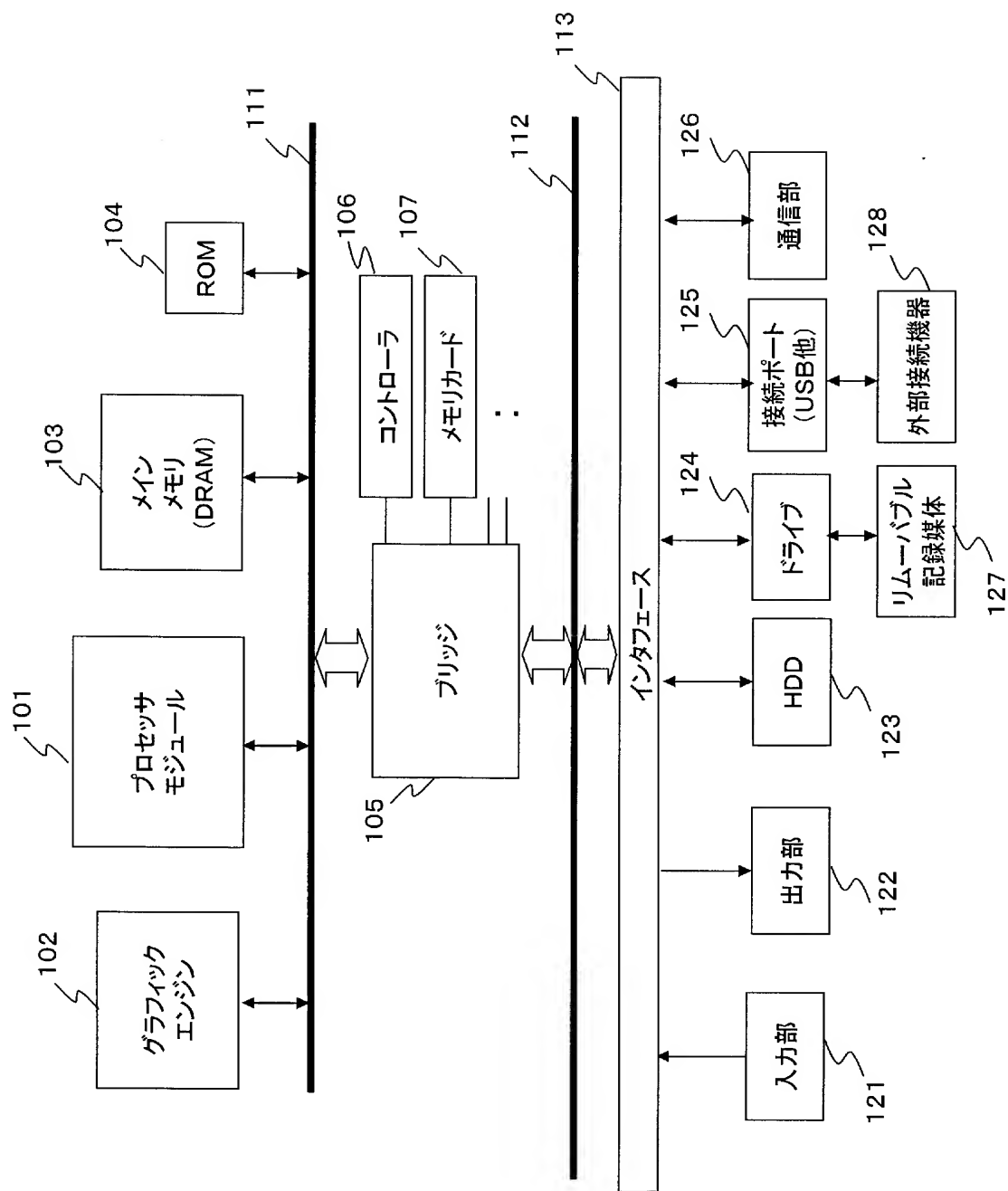
- 101 プロセッサモジュール
- 102 グラフィックエンジン
- 103 メインメモリ(DRAM)
- 104 ROM
- 105 ブリッジ
- 106 コントローラ
- 107 メモリカード
- 111 ホストバス
- 112 外部バス
- 113 インタフェース
- 121 入力部
- 122 出力部
- 123 HDD
- 124 ドライブ
- 125 接続ポート
- 126 通信部
- 127 リムーバブル記録媒体

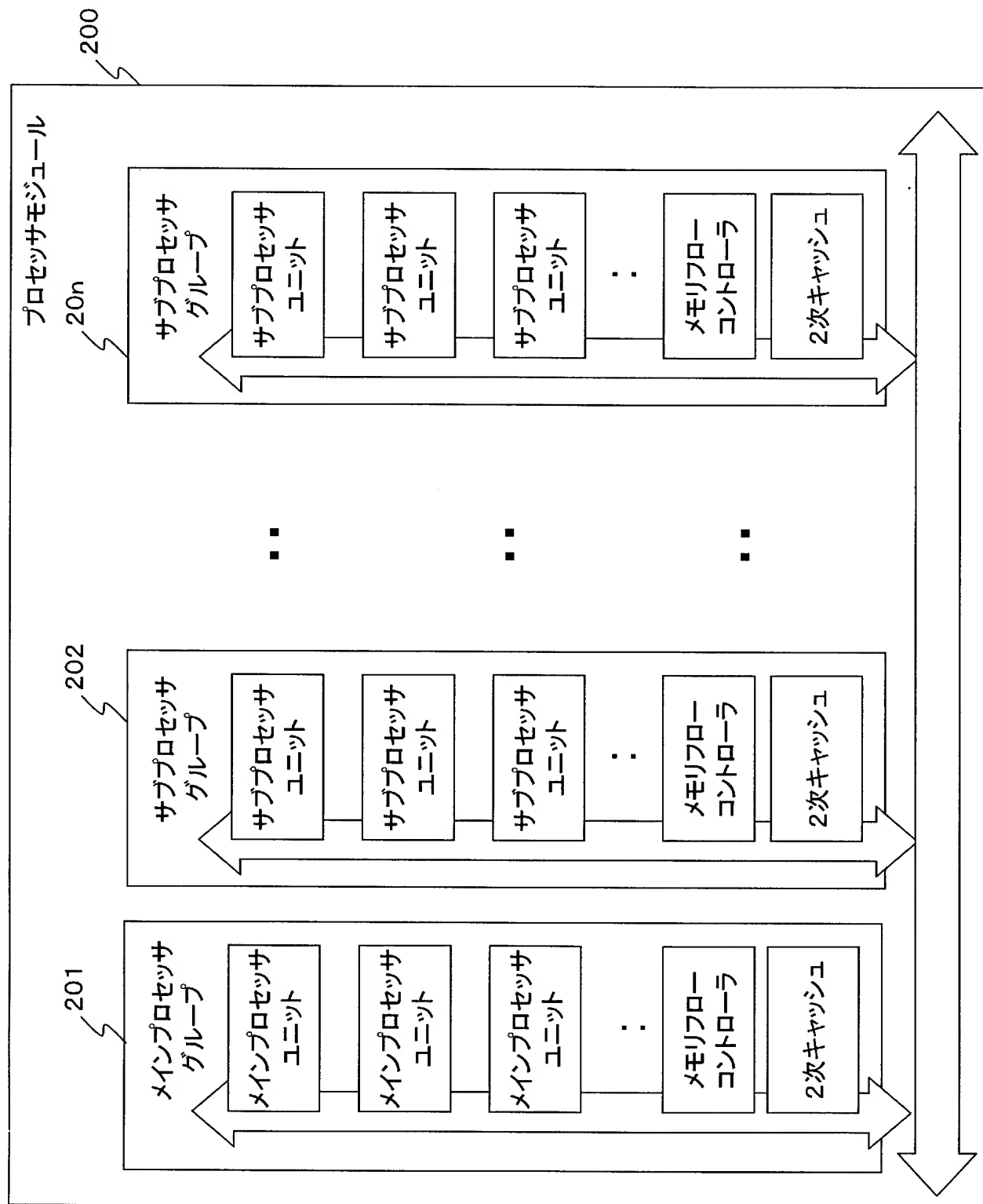
- 1 2 8 外部接続機器
- 2 0 0 プロセッサモジュール
- 2 0 1 メインプロセッサグループ
- 2 0 2 ~ 2 0 n サブプロセッサグループ
- 3 0 1 制御OS
- 3 0 2 , 3 0 3 ゲストOS
- 3 0 4 システム制御OS
- 3 0 5 , 3 0 6 ゲストOSアプリケーション
- 3 0 7 システム制御プログラム
- 4 0 1 メインプロセッサ
- 4 1 1 ~ 4 1 4 物理サブプロセッサ
- 5 0 1 ゲストOSアクセス不可能リソース
- 5 0 2 ゲストOSアクセス可能リソース
- 5 1 0 物理サブプロセッサ
- 5 2 1 レジスタ
- 5 2 2 メインプロセッサから直接アクセスできないI/Oポート
- 5 2 3 メインプロセッサから直接アクセスするI/Oポート
- 5 2 4 ローカルストレージ
- 5 2 5 メインメモリ
- 5 3 1 コンテキストテーブル
- 5 6 0 ゲストOS論理パーティションアドレス空間
- 5 6 1 MMIO+ローカルストレージ領域
- 5 6 2 コンテキストテーブル
- 5 7 0 物理アドレス空間
- 6 1 0 制御OS
- 6 1 1 システムコール処理部
- 6 1 2 論理サブプロセッサスケジューリング処理部
- 6 1 3 コンテキスト管理部
- 6 1 4 メモリ管理部
- 6 2 0 ゲストOS
- 7 0 1 論理サブプロセッサ(ア)



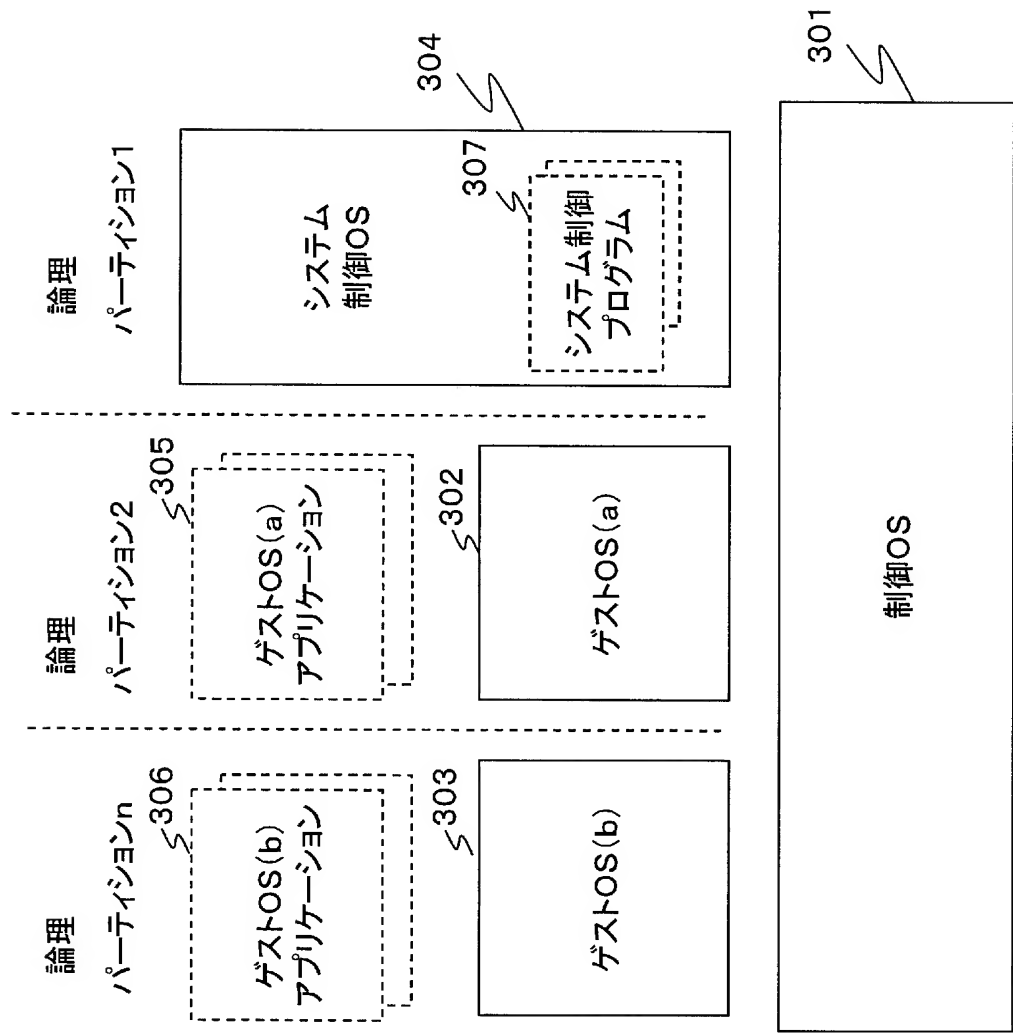


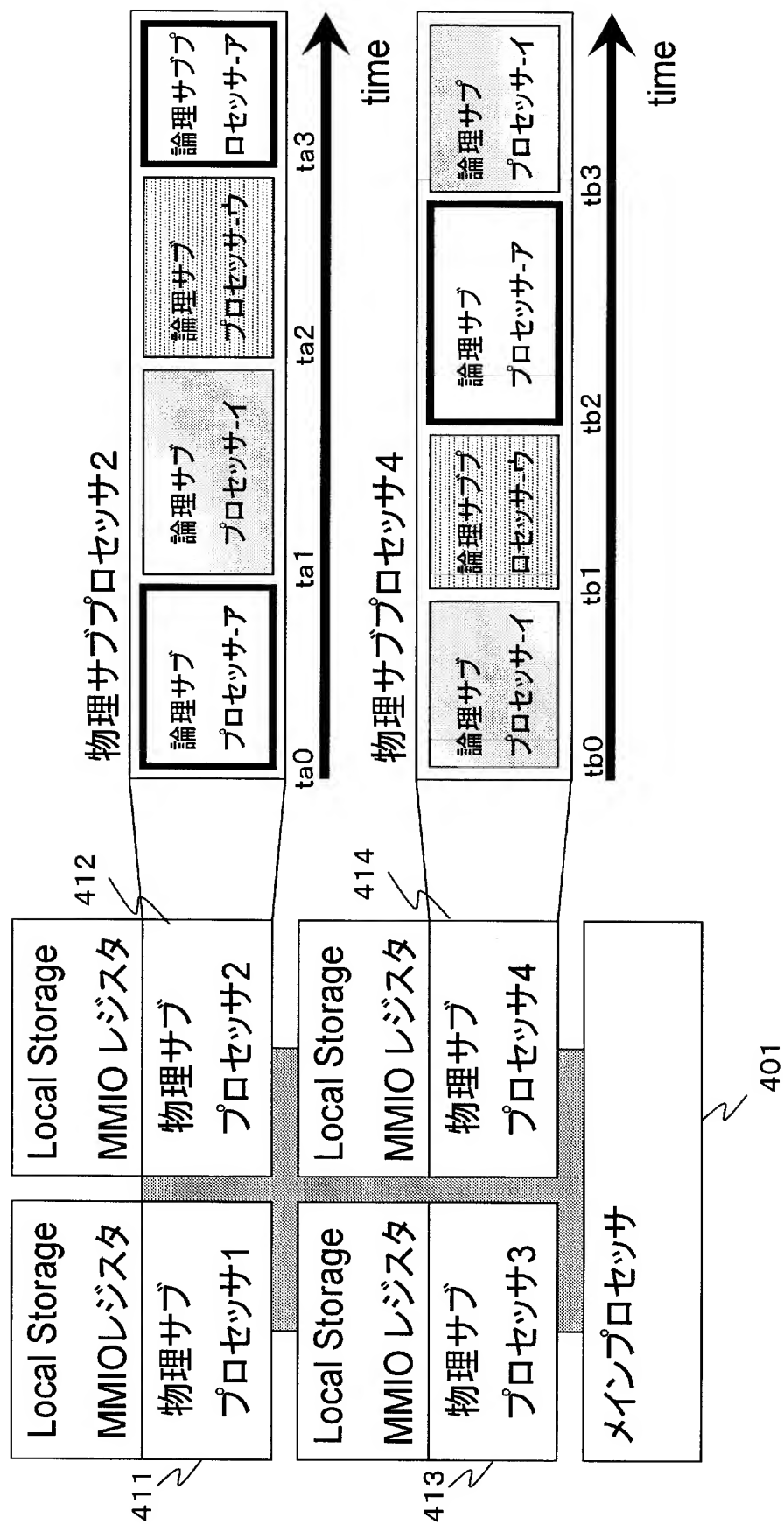


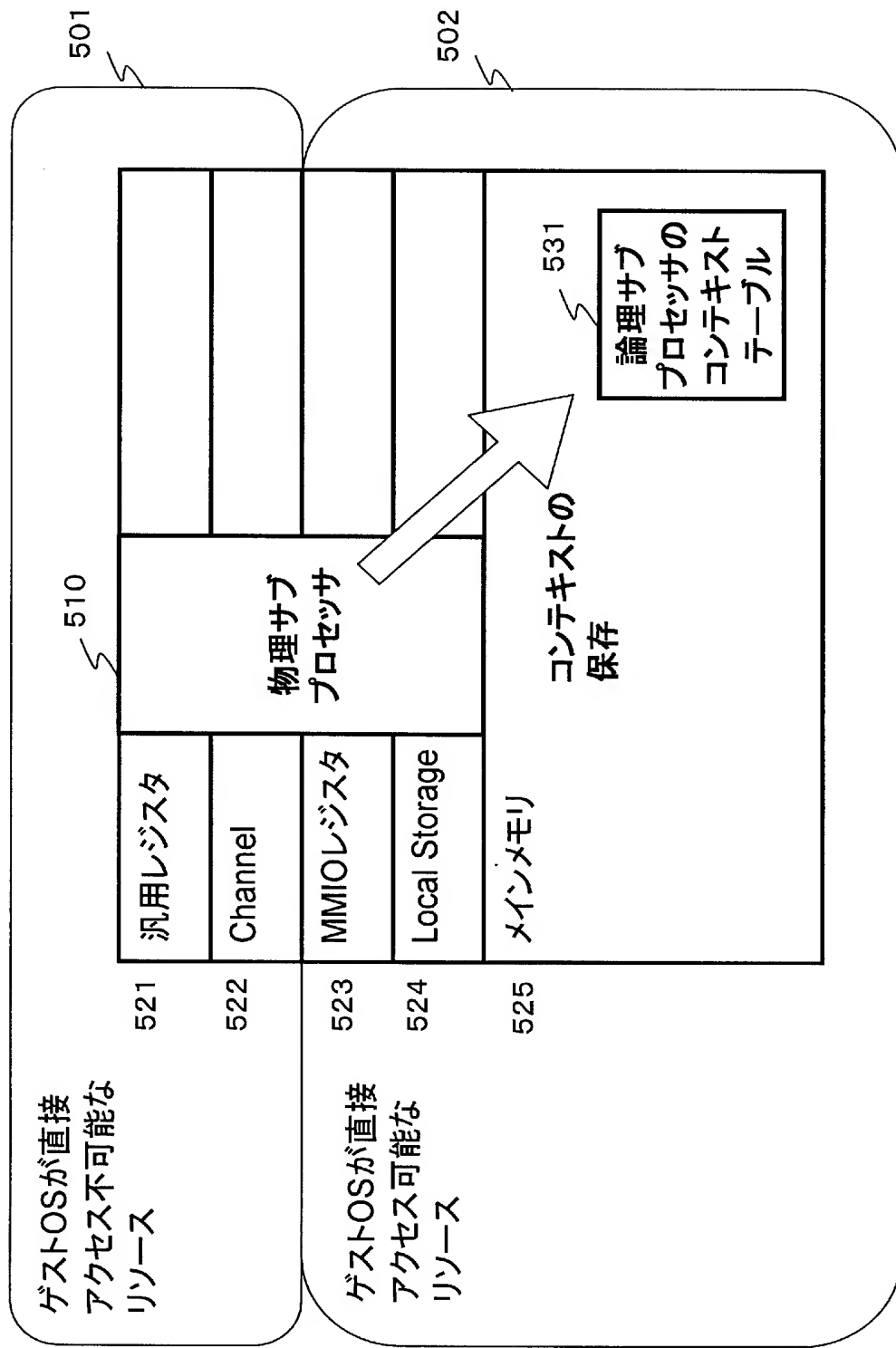


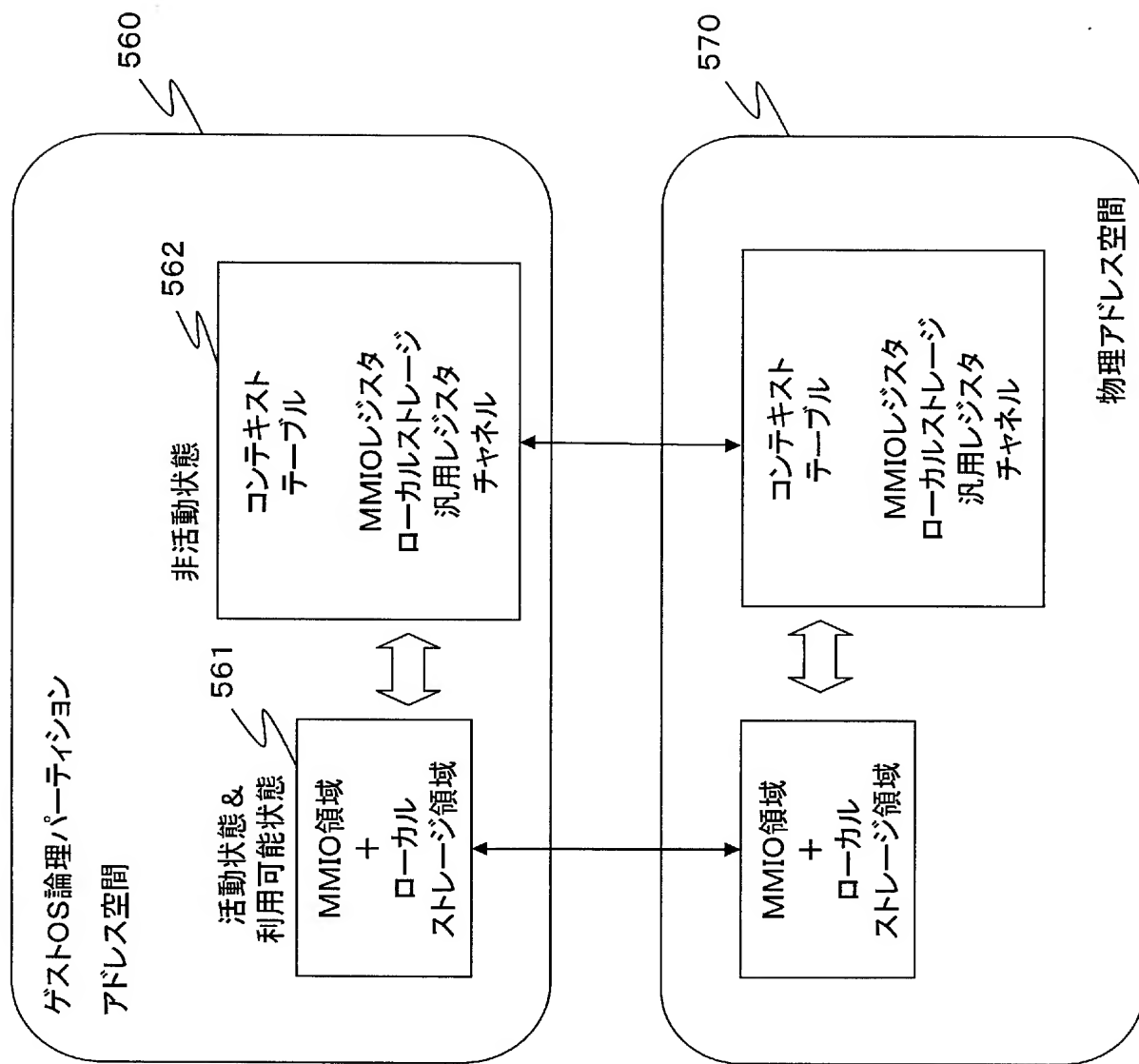


【図 5】

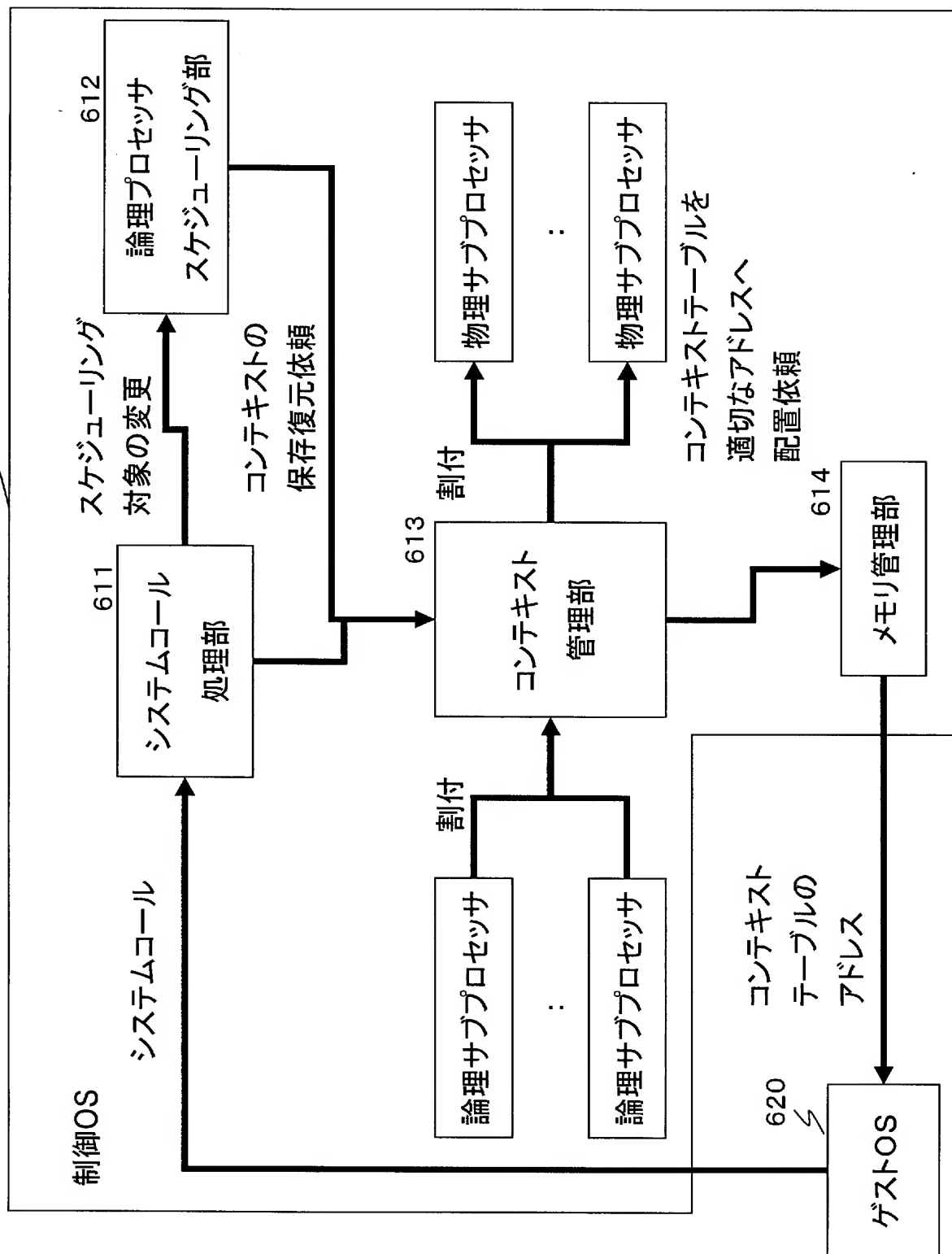


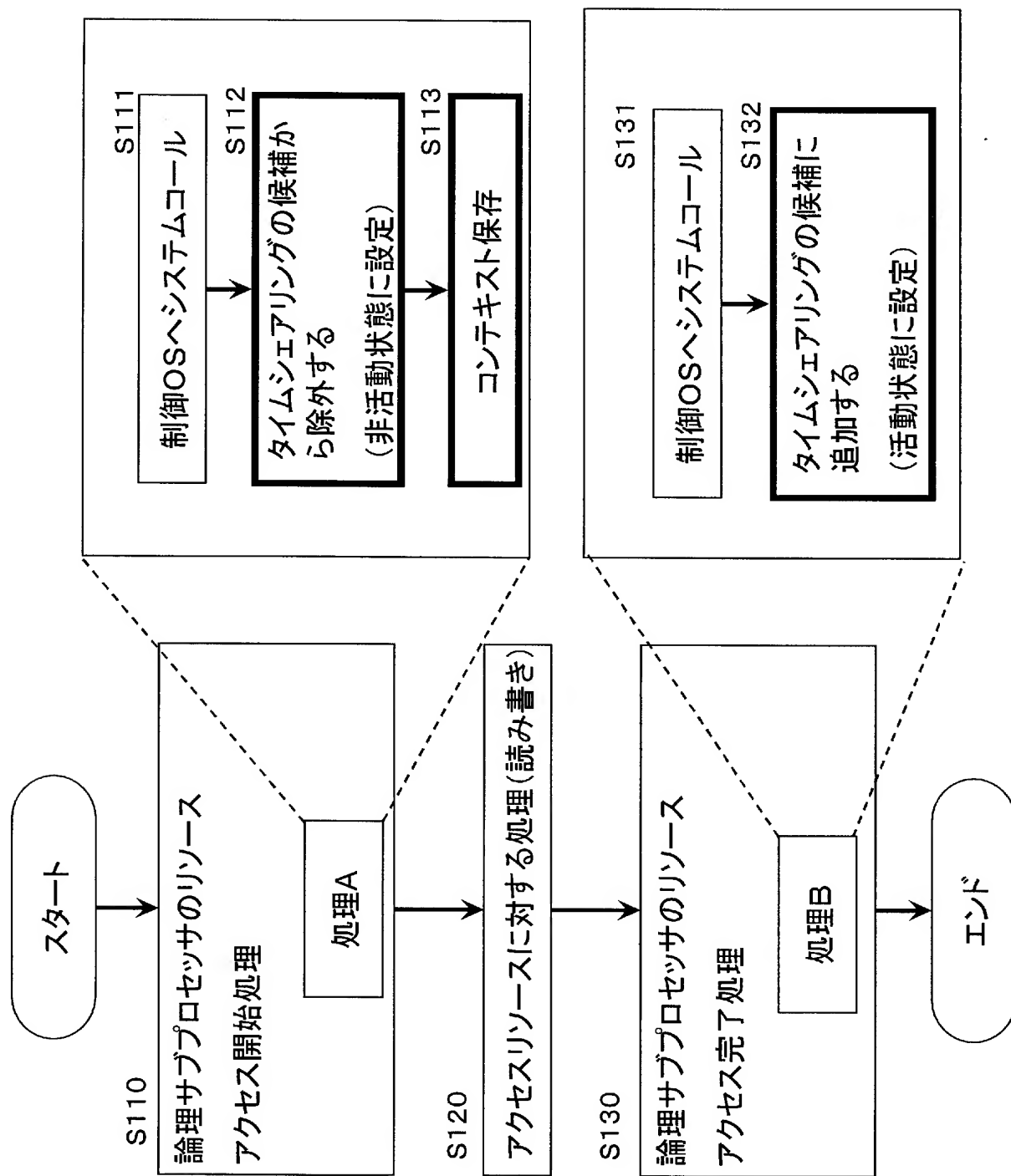




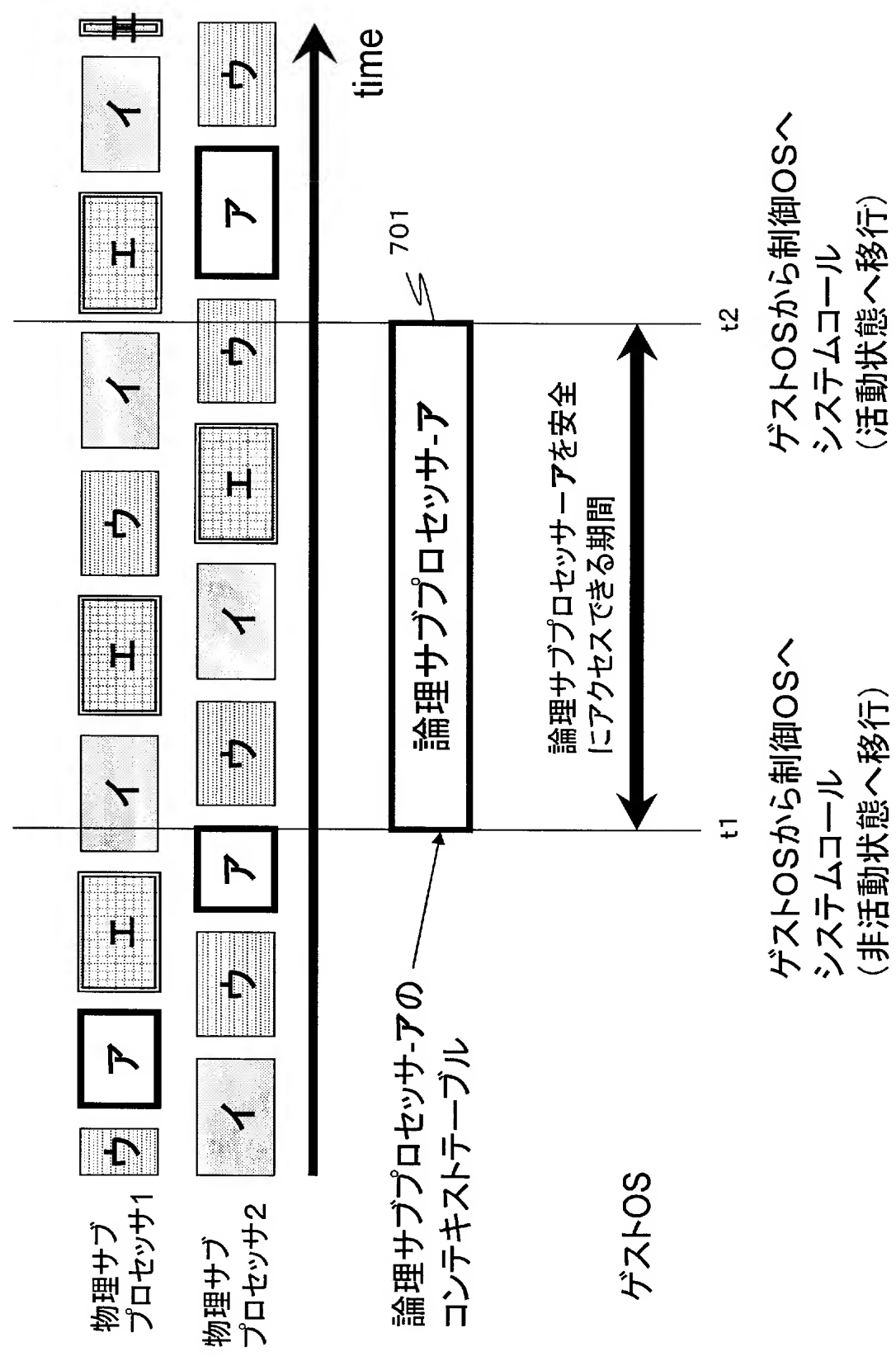


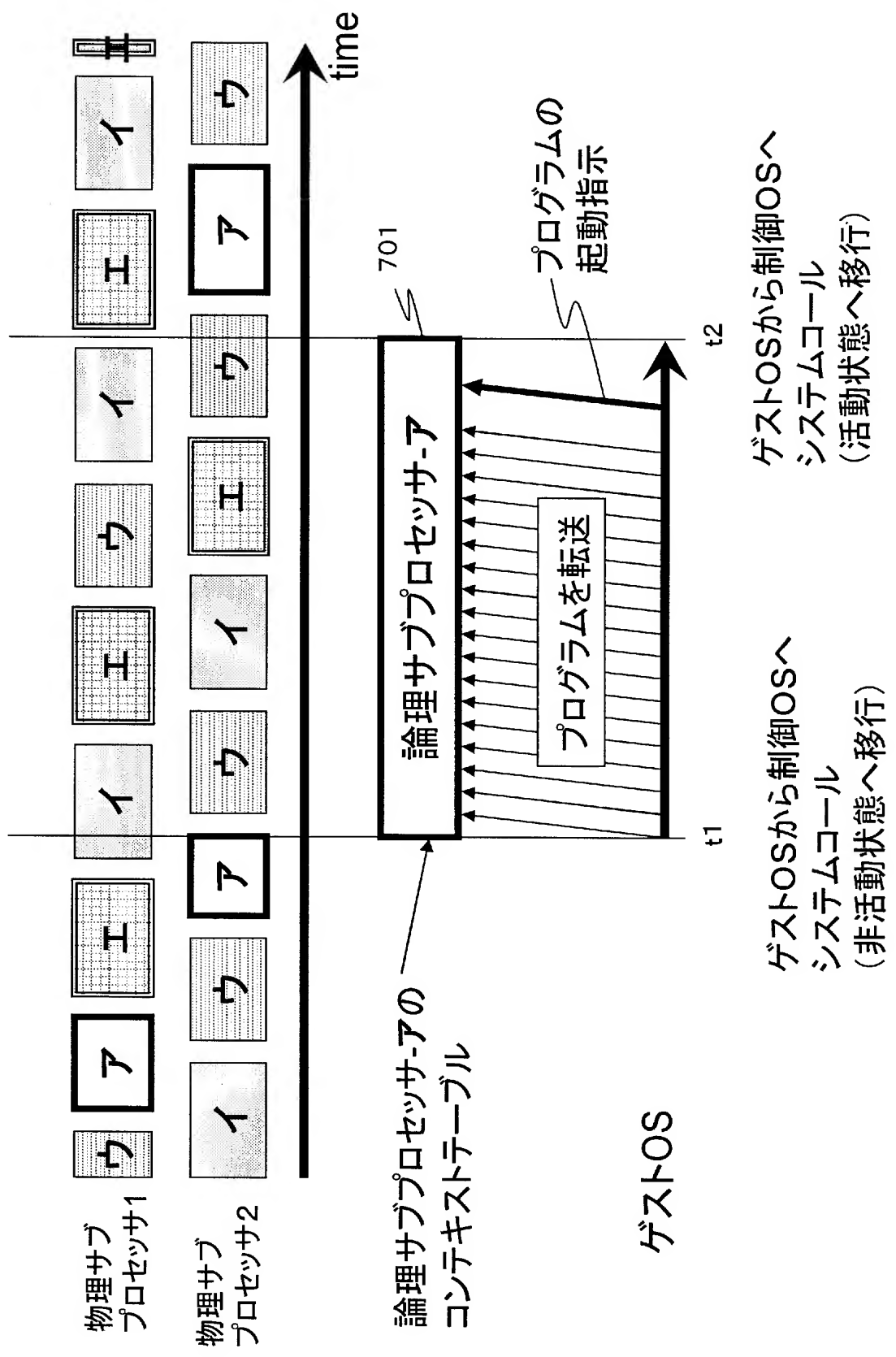
610











【書類名】 要約書

【要約】

【課題】 改良されたコンテキスト保存構成によりデータ処理効率を向上させた装置および方法を提供する。

【解決手段】 複数の論理プロセッサをタイムシェアリングによって物理プロセッサに対応付けてデータ処理を実行する構成において、物理プロセッサの非適用状態にある論理プロセッサ対応のコンテキストテーブルを、論理プロセッサの適用主体としての論理パーティションに対応する論理パーティションアドレス空間にマップして保存する処理を実行する。本構成により、論理プロセッサが物理プロセッサに割り当てられていない状況において、論理プロセッサのコンテキストの取得が可能となり、論理プロセッサに対するアクセス、プログラムロードなどの処理を物理プロセッサ割り当てタイミグを待機することなく実行可能となりデータ処理効率が向上する。

【選択図】 図 9

## 出願人履歴

0 0 0 0 0 2 1 8 5

19900830

新規登録

5 9 7 0 6 2 9 9 3

東京都品川区北品川 6 丁目 7 番 3 5 号

ソニー株式会社